Journal of Informatics and Web Engineering

Vol. 3 No. 1 (February 2024) eISSN: 2821-370X

A Lung Cancer Detection With Pre-Trained CNN Models

Chai Chee Chiet[1], Khoh Wee How[2*], Pang Ying Han[3] and Yap Hui Yen[4]

[1,2,3,4] Faculty of Information Science and Technology, Multimedia University, Malaysia

**corresponding author:(whkhoh@mmu.edu.my；ORCiD: 0000-0002-7338-8427）*

Abstract - Lung cancer is a common cancer in Malaysia, affecting the majority of male citizens. The early detection of lung cancer will decrease its death rate. The only way to detect lung cancer is with a CT scan, and it also requires the doctor to check the scan to confirm the disease. In another way, the computer's support for the detection and diagnosis tool will assist doctors in determining lung cancer more accurately and efficiently. There are three main objectives for this research work. The first target is to study state-of-the-art research work to detect and recognize lung cancer from CT scan images. Then, the article will aim to adopt pre-trained convolutional neural network models in lung cancer detection. It also evaluates the performance of convolutional models on lung cancer imagery data. Then, the pre-trained models with a few added layers and modifications to parameters such as epochs, batch size, optimizer, etc. to conduct model training in this article. After that, Python Pylidc is used in image pre-processing to filter the dataset. Overall, pre-trained models such as ResNet-50, VGG-16, Xception, and MobileNet achieve above-state-of-the-art performance in classifying lung cancer from CT scan images in the range of 78% to 86% accuracy. The best detection accuracy result is the pre-trained VGG-16 model with the addition of some fully connected layers, 16 batch sizes, and the Adam optimizer, which achieved 86.71%.

Keywords— Convolution neural network, Lung Cancer, Image Processing, pre-trained CNN models

Received: 14 June 2023; Accepted: 31 August 2023; Published: 16 February 2024

I. INTRODUCTION

According to the lung cancer statistics collected by Rajadurai et al. [1], the cases happened in a case on every fifty-five Malaysian Males. The higher risk is among the Chinese males with the one in fourth-three persons. The lung cancer cases in Malaysian women are estimated at one in one hundred thirty-five. The statistics data stated that the early detection of lung cancer is critical because the probability to get lung cancer in Malaysian can considered as high chance in over 30 million citizens. Since the early detection of lung cancer is important to save human life, it will be a supportive tool in early detection. The research proposes a model with a deep learning algorithm to detect lung cancer by using the patient CT scan image. This research also focus on finding a way to make the pre-processing of the imagery data easier to make it more understandable for public usage. On the other side, the proposed method uses the dataset that can be accessed publicly or shared by the articles or organizations. Akitoshi Shimazaki et al.'s [2] studies has found out some of the limitations of chest radiography, which are similar to those chest CT in terms of sensitivity and needed to be filtered. So, pre-processing is applied to the CT scan in DICOM format to remove the low-quality CT scan. The concept of a convolutional neural network is applied and explained

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2024.3.1.3

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

in this research. The publicly available concept and current CNN architecture will be used to implement the method, with a few more layers to try to find out which combination of the models provides a better result.

II. LITERATURE REVIEW

In the early stage of lung cancer detection, SVM model is commonly used to detect the patient's lung CT whether it is normal or abnormal. Mahale et al. [3] proposed an SVM classifier with the concept of the CAD system to detect lung cancer disease. Computer-aided design (CAD) is usually used to support the analysis and optimization of an algorithm design. According to this research work, accuracy of Linear Kernel is reported to be obtained 93.023% while the polynomial kernel achieved accuracy of 86.046%. The weakness of this research is that it uses a small dataset, which only contains 30 lung cancer images. These authors did not test the model's performance on larger datasets. Then, the pre-processing of the imagery data in this paper only applied smoothing to reduce the noise of the image.

Furthermore, Makaju et al. [4] found a new solution for lung cancer detection. The authors designed a new model with the SVM for lung cancer detection where this new proposed model had hit an accuracy of 92%, sensitivity of 100%, and specificity of 50%. This proposed research used classification learner toolbox aids for features extraction by developing a trained detection model. In this paper, the authors realised the DICOM format file was difficult to process in the pre-processing stage. The authors decided to use MicroDicom software to convert it into JPEG grayscale images and reduce image noise by implementing the Gabor filter, Median filter, and Gaussian filter. Towards this article, the authors had listed some weaknesses. The first weakness is the image dataset is too huge to download. The second weakness is the classifier can only detect cancer with malignant or benign but cannot classify the lung cancer stages from stage 1 to stage 4.

Nanglia et al. [5] had proposed a model with a combination with SVM and neural networks. The method of this classification algorithm Kernel Attribute Selected Classifier (KASC) is separated into three parts. The first part used BLOCK-PP (Pre-processing Pinelines) for the data pre-processing. The second part represents BLOCK-FEO (Feature Extraction and Optimization) for the feature extraction and optimization while the last part used hybridization of support vector machines (SVM) and neural network for the detection. The authors find that the use of SVM models reduces the complexity of the data and provides a valuable feature set to achieve better classification. The performance result of this KASC hybrid algorithm were reported to achieve the average precision with 98.17%, the accuracy with 98.08%, the recall with 96.5%, and the f-measure with 97%.

Gou et al. [6] have proposed three deep learning models with convolutional neural network, Deep neural network, and stacked autoencoder. The authors used the computer-aided diagnosis (CADx) system to classify lung nodules with benign or malignant. The CADx system is to extract the important features of the images and measure the malignancy by the classifier. The result of the convolution neural network hit the highest accuracy with 84.15%, sensitivity with 83.96%, and specificity with 84.32%.

Sasikala et al. [7] used the CNN model to detect lung cancer for the chest CT images. It extracted the chest CT scans into lung regions in the first stage. Each slice was segmented to get lung tumors. After that, the regions were used to train CNN architecture and test with patient CT images. Inside the segmentation part of this model was also implemented in MATLAB. It trained to be understood and familiarize lung cancer with the sample datasets. This model obtains accuracy with 96%, sensitivity with 87.5%, and specificity with 100%.

Li et al. [8] have proposed a 3D Convolutional Neutral Network with a scheduled learning strategy to classify lung images with the normal condition and cancer lung images. The proposed method has an accuracy of 70.71% with the sequence input order and loss of 56.77%. It also reported an accuracy of 76.44% with the random input order of dataset and loss with 93.31%. Due to the high loss, the authors suggested a sampling choice strategy with the 3D-CNN model to improve accuracy. To counter the high loss result, the authors also suggested not resizing the image somehow to improve accuracy.

On other sides, Kalaivani et al. [9] considered a proposed model with convolutional neural networks. As a result, the model had a hit an accuracy of 90.85%. The authors applied another activation function, ReLU, and a boost function, ADABOOST, in their model training. ADABOOST is used to boost each machine learning algorithm's performance. It is well suited to mix with weak learners, and it is commonly used in the algorithm, a decision tree with only one level. The fact that the authors of this study created a website for their patients to verify the outcomes of their Lung CT pictures is a strength of the study.

Ahmed et al. [10] had investigated a 3D convolutional neural network for early lung cancer detection. The authors focus on exploring whether the lung tissue features can improve the model result. So, the authors aim to remove other substances from the CT scan during the pre-processing stage in order to perform model training. Their model hits the detection accuracy with 80%. According to the model result, it shows that the only lung tissue features for lung cancer detection are not enough to classify the nature of lung cancer.

III. RESEARCH METHODOLOGY

A. Overview System Flow

Figure 1 demonstrates the overview of the proposed system. First, the data from the LIDC-IDRI dataset is used in this research work. The data is pre-processed with a few filter functions like the median filter, anisotropic filter, threshold, and morphology to reduce the image noise and improve the image quality. After the data has been pre-processed, feature extraction is done by using Pylidc in Python to generalize the features with a padding size of 512 and a confidence level of 50% before feeding it into the model to perform model training. The extracted information is saved into metadata. Then, the pre-trained CNN models and the self-proposed CNN models are used to train and test the processed data. The models detect the CT data and categorize the CT images into Ambiguous cases of lung CT, False label for normal lung CT, and True label for cancer cases of lung CT. On the other hand, the machine learning models only use the metadata to do model training for the comparison between the pre-trained models and the self-proposed model.

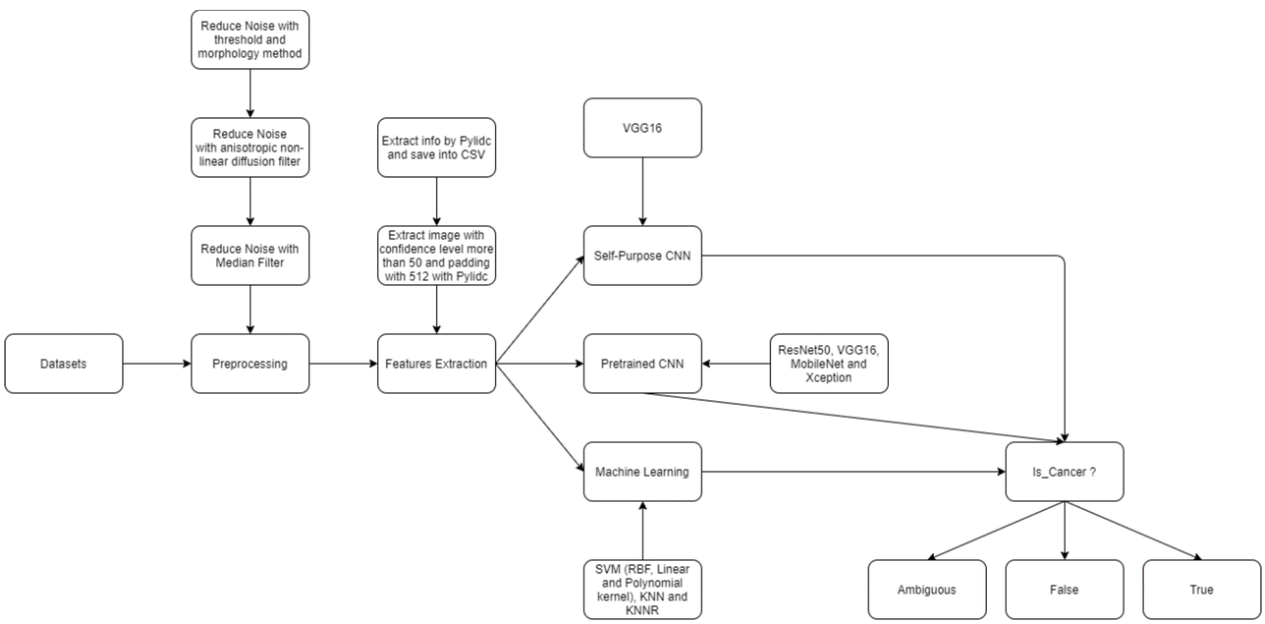

Figure 1. System Flow Diagram

B. Convolutional Neural Network Model

Convolutional Neural Network is a type of deep learning algorithms. It can be used either imagery data or numeric data to assign the nodes with the weights to differentiate the categories of data. One of the benefits of ConvNet is that it does not require many steps in pre-processing as compared with other classification algorithms. The design concept of ConvNet follows the human brain, where neurons connect to each other to build one feature map. The convolutional layer in CNN is to capture the image features with kernel size and movement of pixels according to the number of strides. There are some common types of pooling layers. The first type is Max Pooling, which is used to extract all of the maximum values within the kernel size from the data. The second type of common pooling is Average pooling. It is used to take all the values, using a kernel size, from the data and assign them to an average. A FC (defined as Fully-connected) layer is used to classify the final result of the data categories. The below figure is the concept architecture of CNN, taken from Sumit Saha [11]. Figure 2. shows the concept architecture of the CNN model.

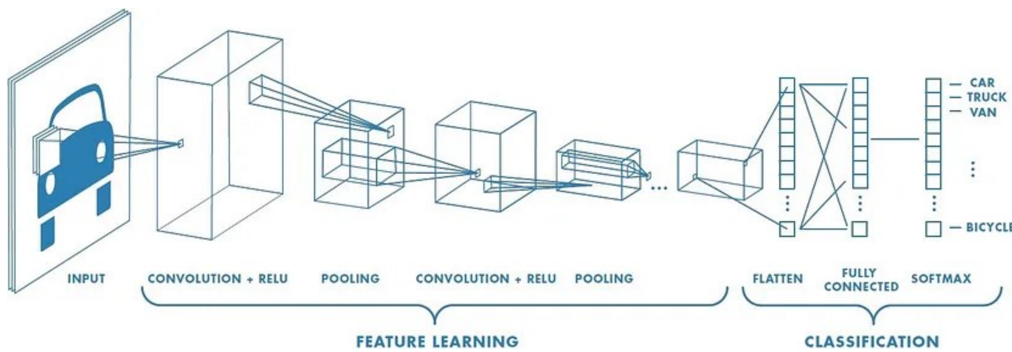

Figure 2. Concept Architecture of CNN Model from Sumit Saha [11]

C. Pre-processing

To make the dataset trainable, a few steps of pre-processing must be done. The first step is to use the Pylidc library to choose the images with a confidence level equal to 50% and make annotations by four doctors. Then, it set a padding size of 512 for the size of the image. The second step checks the nodule annotation on the images. If the nodule annotation is greater than 0, it applies a Pylidc consensus function to count the nodule number and nodule annotation (malignancy). After that, both images with nodule annotations that are greater than 0 and images with no annotations undergo two filter functions to reduce the noise on the DICOM images. The first filter function uses a median filter with a kernel size of 3 to reduce some noise in the original images. After that, it applies anisotropic diffusion [12] as the second filter function to reduce the noise in the images again without blurring the details of the images. In the second step, it reshapes the images by fitting the middle area of the images with KMeans clustering. (x = 100:400, y=100:400 pixels region). Third, a threshold that combines morphology methods with erosion and dilation is used to reduce the noise on the lung segment and make it easier to view the lung wall and lung nodules in the images. Lastly, the images are saved in the .npy format, and the image information, which contains the patient id, patient image id, number of nodules, number of annotations (malignancy) and label of the cancer categories. The two images below show a sample of original lung images in DICOM format (Figure 3.) and a sample of lung images with NPY format that after pre-processing (Figure 4.). Lastly, the trainable images only contain 7374 images.

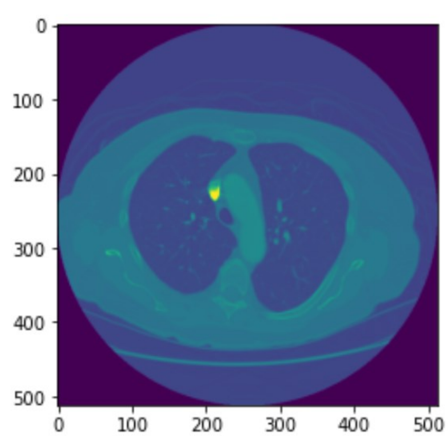

Figure 3. Original Lung CT Image Sample Figure

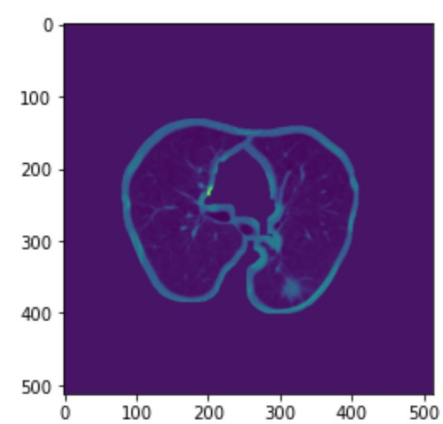

Figure 4. Pre-processed Lung CT Image Sample

D. Proposed Models

This research adopts a few pre-trained CNN models to perform model training for the LIDC datasets, including Xception version 2, MobileNet version 2, ResNet-50, and VGG-16. Since the feature extraction has been done during the pre-processing stage, it uses the pre-processed images to perform the model training. The architecture of the pre-trained models will be discussed in detail in the following section.

In addition, all the pre-trained CNN models use the model weights from ImageNet in Keras applications. After that, it uses the full model architecture and adds in some extra layers. The compilation settings for all pre-trained models include categorical cross entropy for model loss and an Adam optimizer with a 0.0001 learning rate. Then, all pre-trained CNN models are fitted with 16 batch sizes, 30 epochs, and validation split settings of 20% to perform model training. Furthermore, only ResNet-50 uses early stopping and reduction during the model training, and the image features used to perform the model training include lung nodule size and number of lung nodules. All models use pre-processed imagery data (CT scan) as input with a size of 512×512.

i. Xception

The Xception is a convolutional neural network that contains a total of 71 layers with an input of 224×224×3 dimensions. The concept of this model is derived from Francois Chollet [13]. All the separable layers have the same kernel size setting of 3×3. The max-pooling and convolution layers have the same kernel size of 3×3 and stride of 2×2. All the convolution layers and separable convolution layers are followed by a batch normalization [14] that's used to avoid outliers in the training process.

Initially, the entry flow of the model is an input layer with a 224×224×3 input size. The following two layers are the convolution layers, followed by the ReLU activation function and different filter size settings of 32 and 64. After that, it links with the first 1D convolution layers, which contain a separable convolution layer with 128 filter sizes, another activation ReLU function pair with a separable convolution layer with the same 128 filter sizes, and one max-pooling layer. Two sets of 1D convolution layers are added. Both contain a pair of ReLU activation functions and one pair of separable convolution layers with filter sizes of 256 and 728, followed by one max-pooling layer. In the middle flow, it links three sets of the ReLU activation functions with separable convolution layers with 728 filter size by repeating 8 times.

The exit flow of the Xception model is followed by a 1D convolution layer that contains one ReLU activation function with separable convolution layers of 728 filter size, one ReLU activation function with separable convolution layers of 1024 filter size, and one max-pooling layer. It is followed by one separable convolution layer that combines the RELU function with different filter sizes, such as 1536 and 2048. The additional layers that follow are the extra settings from self-modification. The following layers apply a new flatten layer, followed by a fully-connected layer combining the activation function with ReLU, and it is set with a 64-filter size. The second-to-last layer uses a dropout layer with 0.5 settings to prevent model overfitting, and the last layer is a fully connected layer that combines the Softmax activation function with a filter size of 3 to classify the images. Figure 5. shows the proposed design of the pre-trained Xception model architecture of this article.

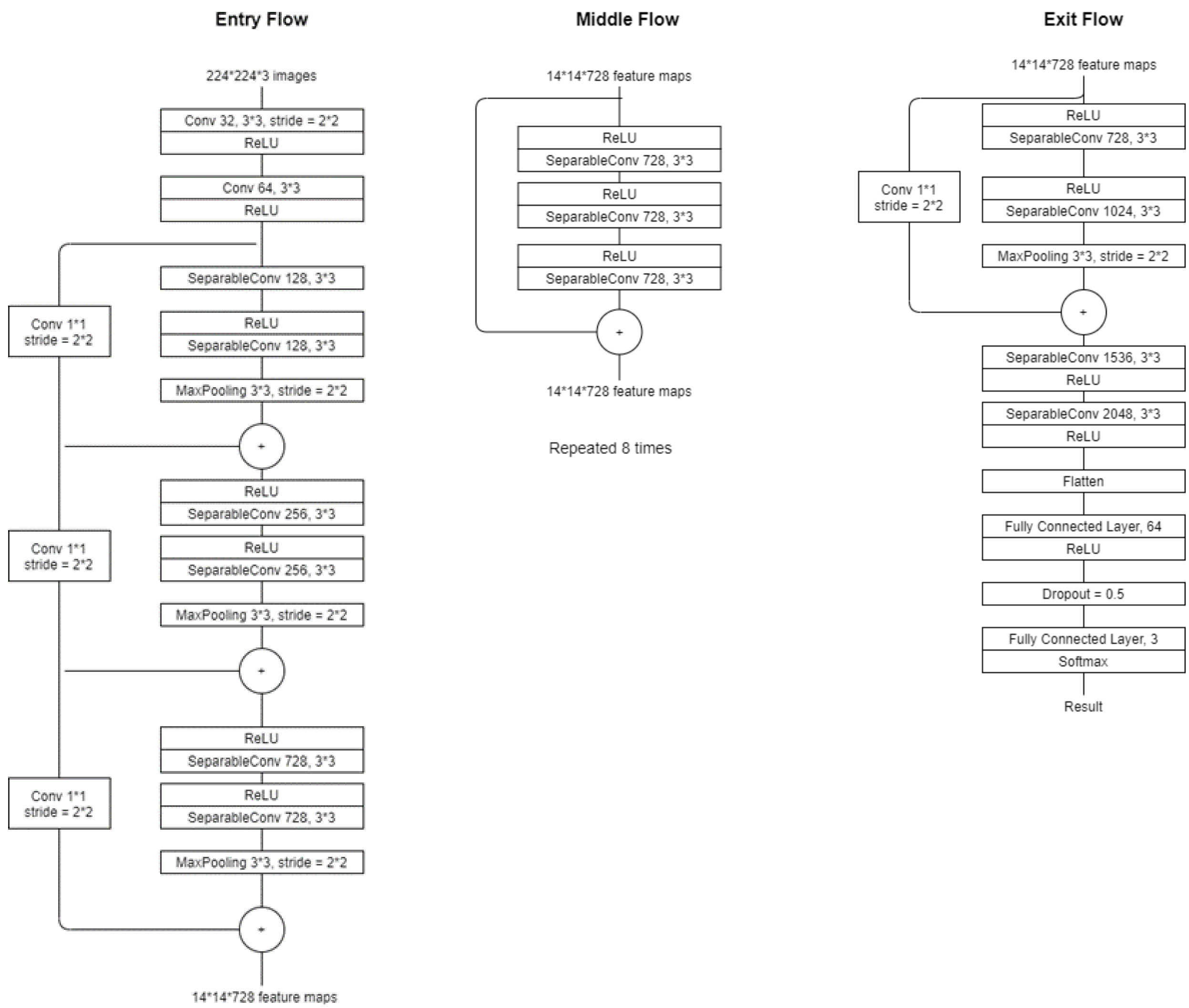

Figure 5. Xception Architecture

ii. MobileNet

The MobileNet's model design concept was proposed by Howard et al. [15] [16]. There are two different types of convolution layers: a depthwise convolution layer and a pointwise convolution layer [17]. The depthwise convolution layer can be considered the channel-wise spatial convolution, and the convolution size is according to the channels. Then, the pointwise convolution is the 1D convolution layer for changing the dimension.

At the end of the model, a flat layer and two fully connected layers will be set up as the additional layers at the exit flow. The first FC layer is set with 64 filter sizes and an activation function with ReLU, while the second FC layer is set with 3 filter sizes and an activation function with Softmax. Both are used to classify the image categories. Figure 6. shows the proposed pre-trained MobileNet model architecture of this article.

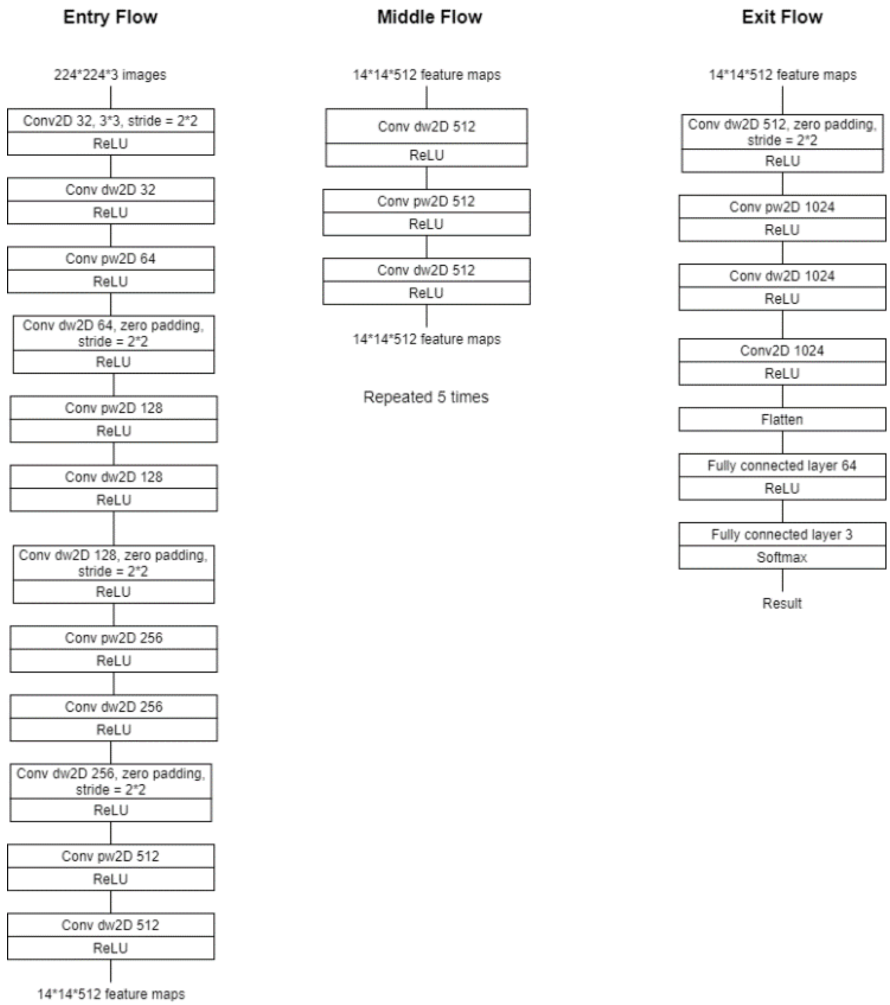

Figure 6. MobileNet Architecture

iii. ResNet-50

The concept architecture of ResNet-50 was proposed by He et al. [18]. All the convolutional 2D layers are followed by one normalization function with batch setup and one activation function with ReLU.

After connecting with "conv5", it applies a flattening and links with three fully connected layers. On the first fully connected layer setting, a 128-filter size and a ReLU activation function are applied. After that, it uses a dropout function with a 50% value. The second fully connected layer has a 64-filter size and an activation function with ReLU. Before the last fully connected layer, it uses the dropout function with a 50% value. Then, the last layer is a classification layer with a 3-filter size and a Softmax activation function. It is used to classify the image result. Figure 7. shows the proposed pre-trained ResNet-50 model architecture of this article.

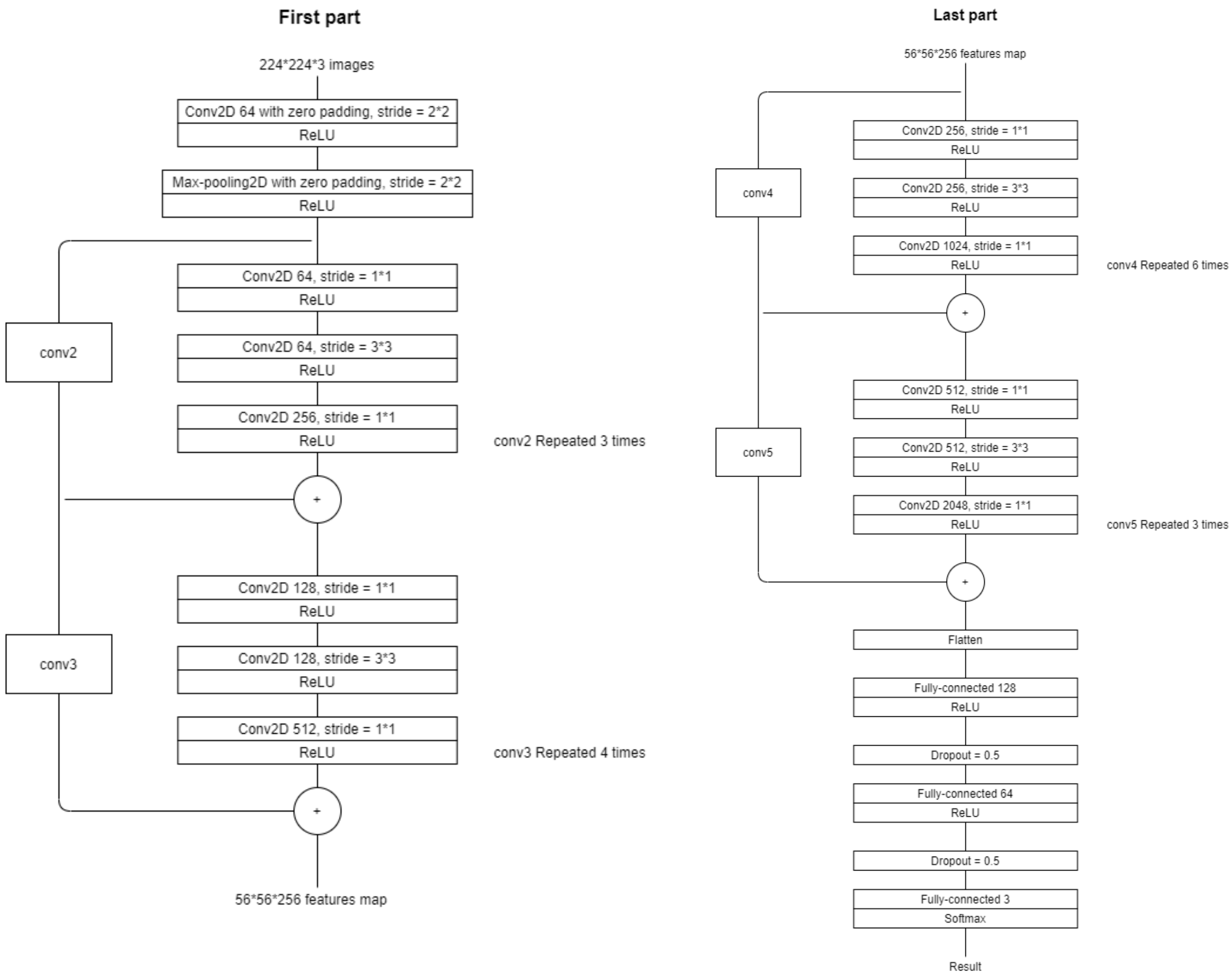

Figure 7. ResNet-50 Architecture

iv. VGG-16

For the VGG-16, it is separated into pre-trained and self-proposed. This pre-trained model design concept is gathered from Simonyan and Zisserman [19].

In the VGG-16 model, the first layer receives an input size of 224×224×3. It links with 5 pairs of convolutional 2D layers. The first pair of 2D convolutional layers is set with 64 filter sizes and a 2D convolutional layer with 64 filter sizes combined with the 2×2 strides max-pooling. The various pairs of 2D convolutional layers are set with the same settings as the first pair of 2D convolutional layers, and the only difference is the second pair is set with 128, 256 filters for the third pair, and 512 filters for the fourth and fifth pairs. Then, it applies a flatten and links with three fully connected layers. For the first fully connected layer, it is set with 128 filter sizes and followed by an activation function with ReLU. The second fully connected layer has the same settings as the first fully connected layer but has a different filter size of 64 filters. The setting of the last fully connected layer is 3 filter sizes, followed by a Softmax activation function for image classification.

For the self-proposed VGG-16 model, the first layer is an input layer, which received an input size of 512×512×1. It links with a 2D convolutional layer with 32 filter sizes and 3×3 strides, then with a max-pooling 2D with 2×2 strides and a flatten is applied. After the flattening, it is connected to two fully connected layers. The setting of the first fully connected layer is 64 filters and a ReLU activation function. Before it is connected to the last fully connected layer, it uses a dropout function equal to 50% to prevent overfitting of the model. The last fully connected layer settings have 3 filter sizes and a Softmax activation function to determine the image classes. Figure 8. shows the architectures of both the pre-trained VGG-16 and the self-proposed VGG-16.

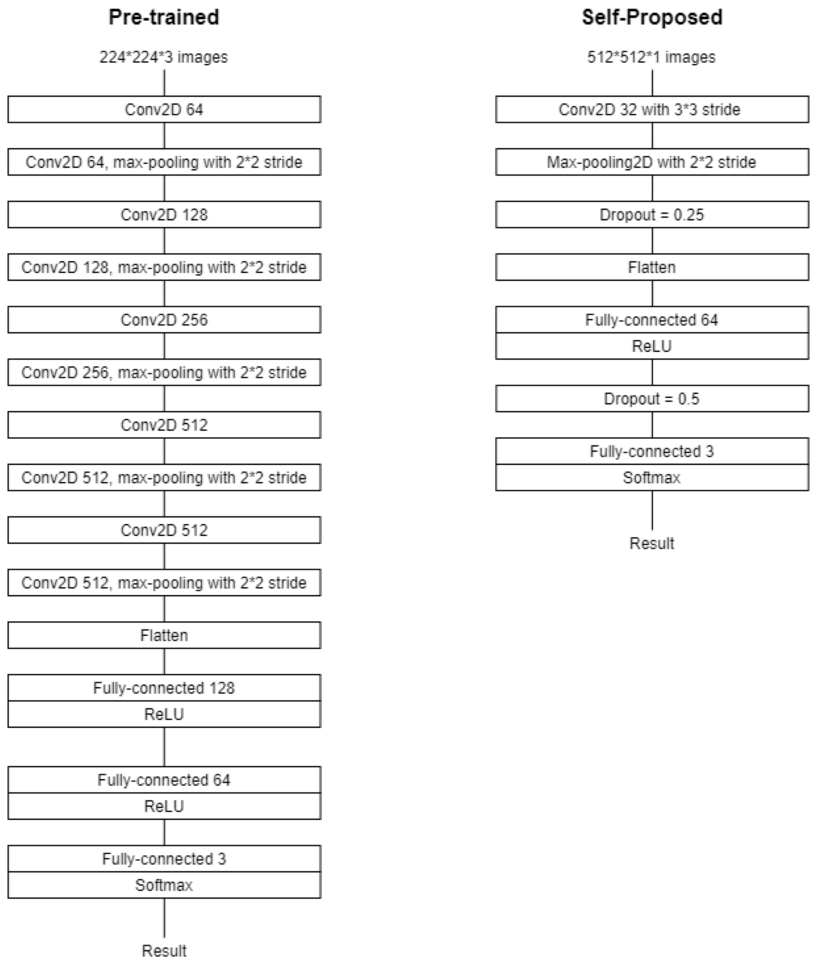

Figure 8. Proposed Pre-trained and Self-Proposed VGG-16 Model Architectures

E. Experimental Settings

i. Dataset

The LIDC-IDRI dataset is a public dataset that is available on the Cancer Imaging Archive website. It allows users to download data with the NBIA Data Retriever. The license of LIDC-IDRI belongs to the Creative Commons Attribution Unported License with version 3. It belongs to the TCIA team. The LIDC-IDRI can be accessed publicly by researchers. The database contains images collected from 301 patient cases with a total of around 53000 DICOM images, and it is the newest database version in 2022. After pre-processing, the trainable images are 7374 images. Figure 9. , Figure 10. , and Figure 11. show examples of the lung cancer CT image from LIDC-IDRI.

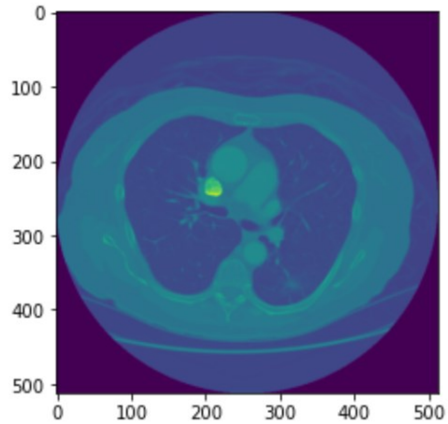

Figure 9. Lung Cancer CT Scan Image

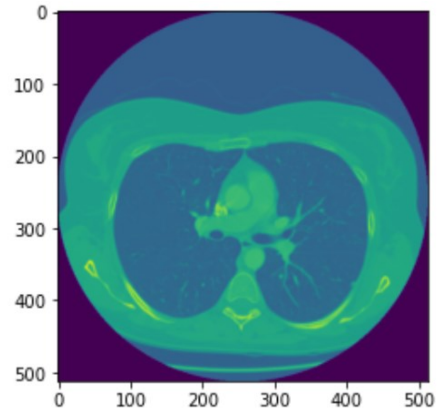

Figure 10. Normal Lung CT Scan Image

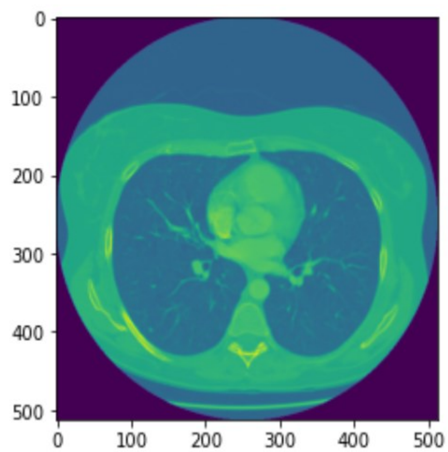

Figure 11. Ambiguous Case of Lung CT Scan Image

The samples for the LIDC-IDRI database are split into subsets for training, validation, and testing as shown in Table 1. Before the model training, it is required to resize the pre-processing imagery data into 224×224×3 by numpy for the pre-trained CNN models and reshape it into 512×512×1 for the self-proposed CNN model. It then uses a labelbinarizer function to convert the labels from string to binary because CNN models cannot use string data to perform training (Ambiguous = 0, Normal = 1, True Cancer = 2). The training and testing subsets for all CNN models are split randomly with 42 random states into 80% (5900) of the imagery data for training and 20% (1474) of it for testing. For the validation of the dataset, it will choose 20% (1180) of the training subset to evaluate the model's performance, as shown in Table 2. Data splitting is required to prevent data redundancy during the model training because some imagery data may look similar to each other.

Table 1. Details of LIDC-IDRI.

Image Size	Normal	Cancer	Ambiguous	Total
512*512	4441	1698	1235	7374

Table 2. Data Splitting Details

Class	Samples
Training	5900
Validating	1474
Testing	1180

IV. RESULTS AND DISCUSSIONS

Figure 12 shows a comparison of all models' accuracy results. Table 3 shows all the models' performance summaries. It was noticed that the pre-trained model with the highest performance for CNN models, which is VGG-16, hit the accuracy mark with 86.71%. Onwards, the highest performing machine learning models are KNN regression, SVM with a linear kernel, and SVM with an RBF kernel, and all of them hit 100% accuracy.

According to the pre-trained CNN models results, the models that contain more convolutional layers with a max-pooling function somehow achieve better accuracy results. It is because it gets more features or more parameters during the model's training. The model with batch normalization can produce better training results because it can avoid outliers. Another factor contributing to the low accuracy is the lack of a dataset and computational power. Both factors prevent most pre-trained CNN models from achieving higher accuracy. A way to boost the model training result is to collect more datasets because it can prevent data imbalances from happening. When the dataset is increasing, the hyperparameter tuning method can be applied for model training; otherwise, it cannot produce an accurate result on small datasets. Then, the device's GPU processing power is also a factor that prevents the model from achieving higher accuracy. A GPU with more computational power and memory can produce better results. It tends to run more epochs for model training and learn better features from the various datasets. Furthermore, as the imagery data has been converted into numerical data, the machine learning model can produce higher accuracy due to the special library function, which can directly extract the important features from the numerical data. The complexity of the imagery data is different as compared to the numerical data.

Table 3 details the AUC, recall, accuracy, and F1-score achieved by all models. This table clearly shows that three machine learning models, including KNNR, SVM with a linear kernel, and SVM with a RBF kernel, achieved 100% AUC. VGG-16 has the best pre-trained CNN model performance, with an AUC of 86.71%.

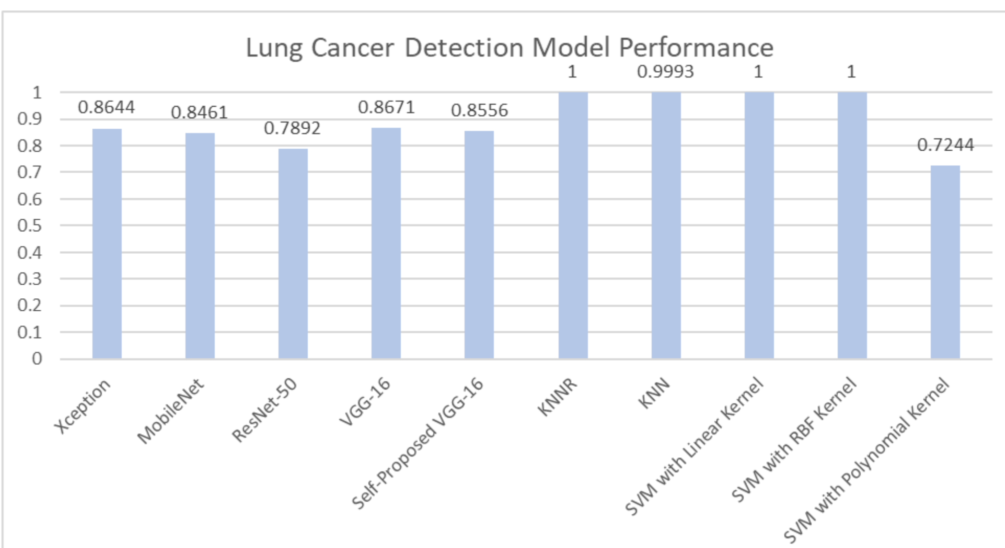

Figure 1. Comparison of the model

Table 3. Performance of the model

Model	Class	Precision	Recall	F-Features	Accuracy
Xception	Ambiguous	0.8547	0.5698	0.6838	0.8644
	Normal	0.8891	0.9542	0.9205	
	Abnormal	0.8055	0.8571	0.8305	
MobileNet	Ambiguous	0.6992	0.7209	0.7099	0.8461
	Normal	0.8863	0.9279	0.9066	
	Abnormal	0.8537	0.7318	0.7881	
ResNet-50	Ambiguous	0.3178	0.8632	0.4646	0.7892
	Normal	0.8581	0.9199	0.8879	
	Abnormal	0.6275	0.8105	0.7074	
VGG-16	Ambiguous	0.7500	0.6977	0.7229	0.8671
	Normal	0.9082	0.9394	0.9235	
	Abnormal	0.8399	0.8105	0.8249	
Self-Proposed VGG-16	Ambiguous	0.8462	0.6395	0.7285	0.8556
	Normal	0.8571	0.9611	0.9061	
	Abnormal	0.8567	0.7493	0.7994	
KNN	Ambiguous	1	1	1	1
	Normal	1	1	1	
	Abnormal	1	1	1	
KNNR	Ambiguous	1	1	1	0.9993
	Normal	1	0.9989	0.9994	
	Abnormal	1	1	1	
SVM-Linear	Ambiguous	1	1	1	1
	Normal	1	1	1	
	Abnormal	1	1	1	
SVM-RBF	Ambiguous	1	1	1	1
	Normal	1	1	1	
	Abnormal	1	1	1	
SVM-Polynomial	Ambiguous	0.2391	0.0213	0.0391	0.7244
	Normal	0.9925	0.8257	0.9014	
	Abnormal	0.4685	0.9971	0.6375	

Figure 2. shows the confusion matrices for the VGG-16 model, which has the best performance.

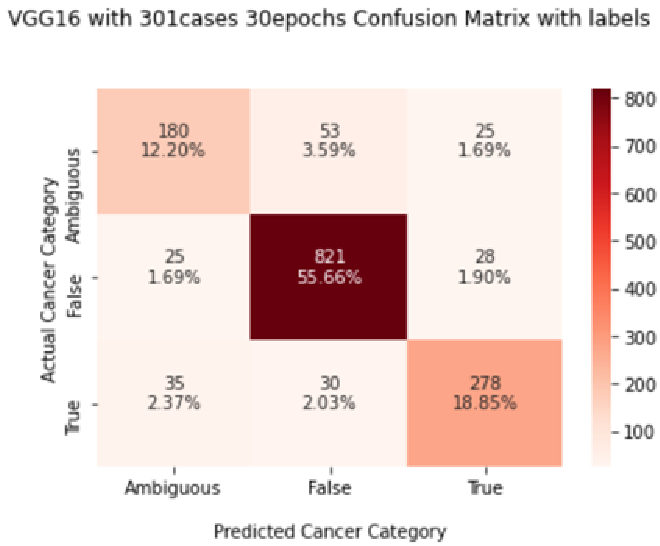

Figure 2. Confusion Matrix of VGG-16 model

V. CONCLUSION

In the findings, four types of pre-trained convolutional neural network models and a self-proposed VGG-16 model have been proposed for the detection of normal, abnormal, and ambiguous lung CT images. The models chosen in this study include VGG-16, ResNet-50, Xception-V2, and MobileNet-V2 for both feature learning and classification. This study uses the pre-processed CSV metadata to conduct the experiments with the machine learning models. The K-Nearest Neighbour Regression (KNNR) and support vector machines (SVM) are used to conduct feature learning and classification on the lung CT categories. The dataset used in this study was obtained from LIDC-IDRI, which is a public, free-source database for research.

The pre-trained CNN model with the highest accuracy for the detection of lung CT scan images is the pre-trained VGG-16. The three machine learning models with the best performance were KNN regression, SVM with a linear kernel, and SVM with a RBF kernel; all of them achieved 100% accuracy. But the accuracy of the detection by using imagery data must be improved in the future due to the life issue. This is because most of the time the data for detection is based on lung CT scan data, and early detection of lung cancer will somehow increase the recovery chance. It may reduce the patient's death rate if early treatment action is taken.

In the future, the DenseNet model is recommended to carry out the experiment because it can train with more parameters and be compared with various CNN models in this paper. Onwards, this research recommends applying a mixture of the metadata and pre-processed imagery data to the CNN model training, which might produce more accurate results.

ACKNOWLEDGEMENT

A version of this paper was presented at the third International Conference on Computer, Information Technology and Intelligent Computing, CITIC 2023, held in Malaysia on 26th-28th July 2023.

REFERENCES

[1] P. Rajadurai, S. Hin How, C. Kim Liam, A. Sachithanandan, S. Yang Soon and L. Mun Tho, "Lung Cancer in Malaysia," *Editorial: Lung Cancer Worldwide,* vol. 15, pp. 317-323, 2020.

[2] A. Shimazaki, D. Ueda, A. Choppin, A. Yamamoto, T. Honjo, Y. Shimahara and Y. Miki, "Deep learning-based algorithm for lung cancer detection on chest radiographs using the segmentation method," *Scientific Reports,* vol. 12, no. 2045-2322, p. 727, 2022.

[3] A. Mahale, C. Rawool, D. Tolani, D. Bathija and K. Jewani, "). SVM classifier based CAD system for Lung Cancer Detection," *ijecs,* 2017.

[4] S. Makaju, P. Prasad, A. Alsadoon, A. Singh and A. Elchouemi, "Lung Cancer Detection using CT Scan Images," *Procedia Computer Science,* vol. 125, no. 1877-0509, pp. 107-114, 2018.

[5] P. Nanglia, S. Kumar, A. N. Mahajan, P. Singh and D. Rathee, "A hybrid algorithm for lung cancer classification using SVM and Neural Networks," *ICT Express,* vol. 7, no. 2405-9595, pp. 335-341, 2021.

[6] G. Jun Feng, S. QingZeng, L. Z., X. L. and D. XueChen, "Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images," *Journal of Healthcare Engineering,* p. 7, 2017.

[7] S. Sasikala, M. Bharathi and B. R. Sowmiya, "). Lung Cancer Detection and Classification Using Deep CNN," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 8, no. 2278-3075, 2018.

[8] Y. Li, Y. Tian and B. Ge, "Lung Cancer Classification Using 3D-CNN with a Scheduled Learning Strategy," *Advances in Intelligent System,* pp. 162-164, 2018.

[9] N. Kalaivani, N. Manimaran, D. Sophia and D. D. D, "Deep Learning Based Lung Cancer Detection and Classification," *IOP P Conf. Series: Materials Science and Engineering 994,* 2020.

[10] T. Ahmed, S. Mst. Parvin, M. Reduanul Haque and M. Shorif Uddin, "Lung Cancer Detection Using CT Image Based on 3D Convolutional Neural Network," *Journal of Computer and Communications,* vol. 8, pp. 35-42, 2020.

[11] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way. Towards Data Science," 2018.

[12] C. A. Palma, F. A. Cappabianco, J. S. Ide and P. A. Miranda, "Anisotropic Diffusion Filtering Operation and Limitations - Magnetic Resonance Imaging Evaluation," *IFAC Proceedings Volumes,* vol. 47, no. 3, pp. 3887-3892, 2014.

[13] F. Chollet, "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)," no. 1251-1258, 2017.

[14] J. Brownlee, "A Gentle Introduction to Batch Normalization for Deep Neural Networks," 2019.

[15] K. S. Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015.

[16] W. Chi Feng, "A Basic Introduction to Separable Convolutions," 2018.

[17] H. Kaiming, Z. Xiangyu, R. Shaoqing and S. Jian, "Deep Residual Learning for Image Recognition," *CoRR,* vol. abs/1512.03385, 2015.

[18] A. G. Howard, Z. Menglong, B. Chen, K. Dmitry, W. Weijun, W. Tobias, A. Marco and A. Hartwig,"MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *CoRR,* vol. abs/1704.04861, 2017.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," ArXiv. /abs/1409.1556, 2015.